TECNOLÓGICO DE ESTUDIOS SUPERIORES DE ECATEPEC

DIVISIÓN DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

BASE DE DATOS PARA DISPOSITIVOS MÓVILES

CRUD Y RUTAS

PROYECTO: COMERCIALIZACIÓN

MÓDULO SELECCIONADO: CRUD DE PRODUCTOS

GRUPO: 5851

PRESENTA:

LÓPEZ ALCÁNTARA ALAN JESÚS

PROFESORA:

CORTES BARRERA GRISELDA

JULIO 02, 2022

CRUD Y RUTAS DEL MÚDULO

ESPECIFICACIONES:

Proyecto: Comercialización

Integrante: López Alcántara Alan Jesús

Módulo: CRUD de productos.

Repositorio Backend: https://github.com/alan11-wq/Backend_Comercializacion

NOTA: Trabajo el proyecto individualmente y por indicaciones de la profesora seleccioné un módulo independiente a trabajar en lugar de trabajar todo el proyecto en completo.

INSTRUCCIONES:

De acuerdo con el proyecto asignado, describe los detalles para construir el Backend del proyecto con las diferentes rutas probadas en POSTMAN para las peticiones al servidor (get, post, put y delete) de acuerdo al módulo asignado por el líder del proyecto.

Así mismo hacer la conexión de la base de datos y mostrar en POSTMAN la pestaña por cada ruta creada para el CRUD de la tabla o tablas correspondientes del módulo asignado.

DESARROLLO:

MÓDULO PRODUCTOS

ARCHIVO DE CONEXIÓN A BASE DE DATOS

```
const mysql = require ('mysql');
const conecta =  mysql.createConnection({
    host:"127.0.0.1",
    user:"root",
    password:"MessiThiago11",
    port:"3306",
    database:"proyectobd"
});
conecta.connect();

module.exports = conecta;
```

ASIGNACIÓN DE RUTAS

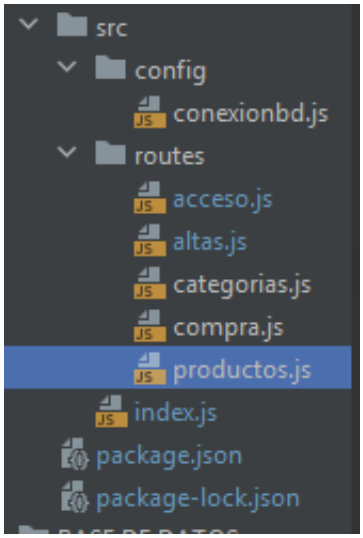
Para esto es importante inicializar el proyecto y determinar una ruta principal del módulo en el archivo index del servidor. En este caso enfocándonos en el módulo de productos vamos a crear la ruta /productos en el archivo index.

```
const express = require ('express');
const app = express();
const port = ('port', process.env.PORT || 5000);

app.set('port', port);
```

```
app.listen(app.set('port'));

const cors = require('cors');
app.use(express.json());
app.use(cors());

app.use('/', require('./routes/acceso'))
app.use('/categorias', require('./routes/categorias'))
app.use('/compra', require('./routes/compra'))
app.use('/alta', require('./routes/altas'))
app.use('/productos', require ('./routes/productos'))

console.log('SERVIDOR EN PUERTO');
```

La ruta marcada nos servirá como una ruta raíz para realizar cada una de las peticiones necesarias para el CRUD.

Posterior a nuestra ruta raíz, definimos un archivo JS el cual contendrá cada una de las peticiones a la base de datos. La estructura quedará de la siguiente manera.



COMPROBACIÓN DE PARÁMETROS EN LA BASE DE DATOS

Ahora es importante identificar la tabla necesaria a la que se realizarán las peticiones en la base de datos. En el caso específico de mi módulo, la información la extraeremos de la tabla de (Producto). En ella nos basaremos para identificar los campos y la información que podremos extraer, modificar, insertar o eliminar.

```
68
69  CREATE TABLE Producto(
70  id_producto int primary key not null,
71  nombre_producto varchar (30),
72  descripcion_producto text (50),
73  precio_producto varchar (30),
74  stock_producto int,
75  id_categoria int,
76  foreign key (id_categoria) references Categoria(id_categoria)
77  );
78
```

Estado original de la tabla Productos.

```
192
193     SELECT * FROM Usuarios
194     SELECT * FROM Tienda
195     SELECT * FROM Producto
196     SELECT * FROM Roles
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id_producto	nombre_producto	descripcion_producto	precio_producto	stock_producto	id_categoria
801	Televisor	Televisor de 70 pulgadas	$1000	10	705
802	Audifonos Sony	Audifonos alambricos Sony	$300	20	701
803	Play Station 5	Consola de Play Station 500gb	$1800	8	701
804	Telefono inalambrico	Telefono inalambrico	$400	20	704
805	Xbox one 1 TB	Consola de videojuegos xbox one	$4999	10	701
806	Xbox one 1 TB	Consola de videojuegos xbox one	$4999	10	701
NULL	NULL	NULL	NULL	NULL	NULL

CRUD

Para la comprobación del CRUD, vamos a crear un nuevo producto, modificaremos un producto que baja su precio y stock, consultaremos ese producto en descuento y eliminaremos un producto repetido en la base de datos.

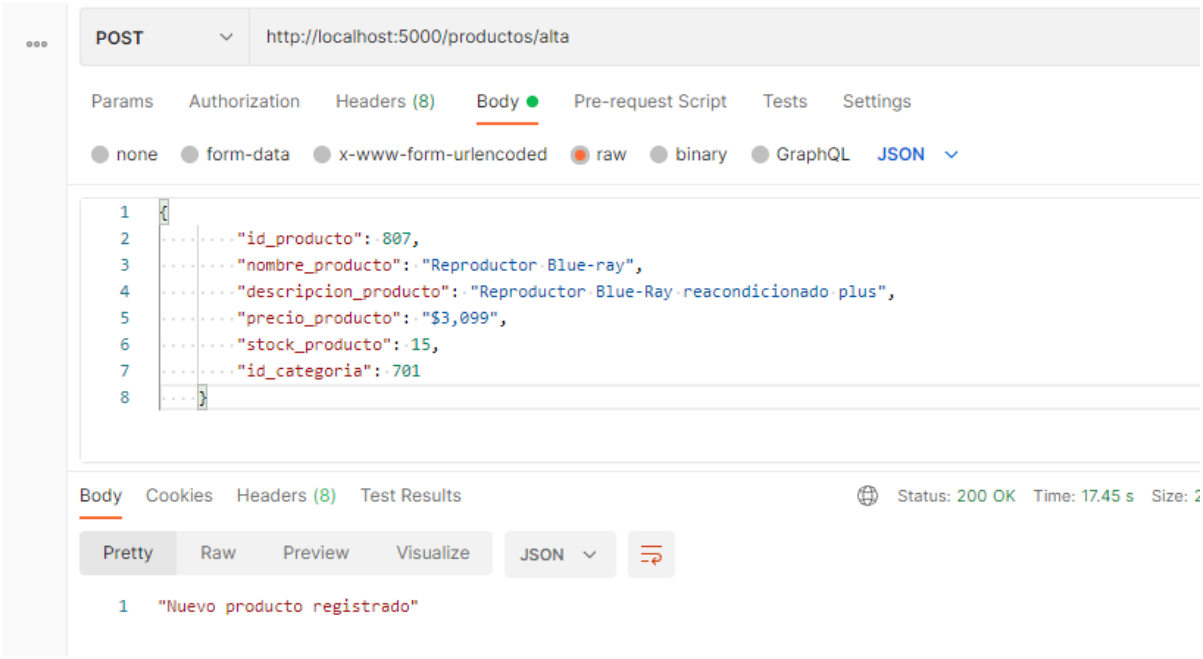
ALTA DE UN PRODUCTO (POST)

Comenzamos dando de alta un nuevo producto.

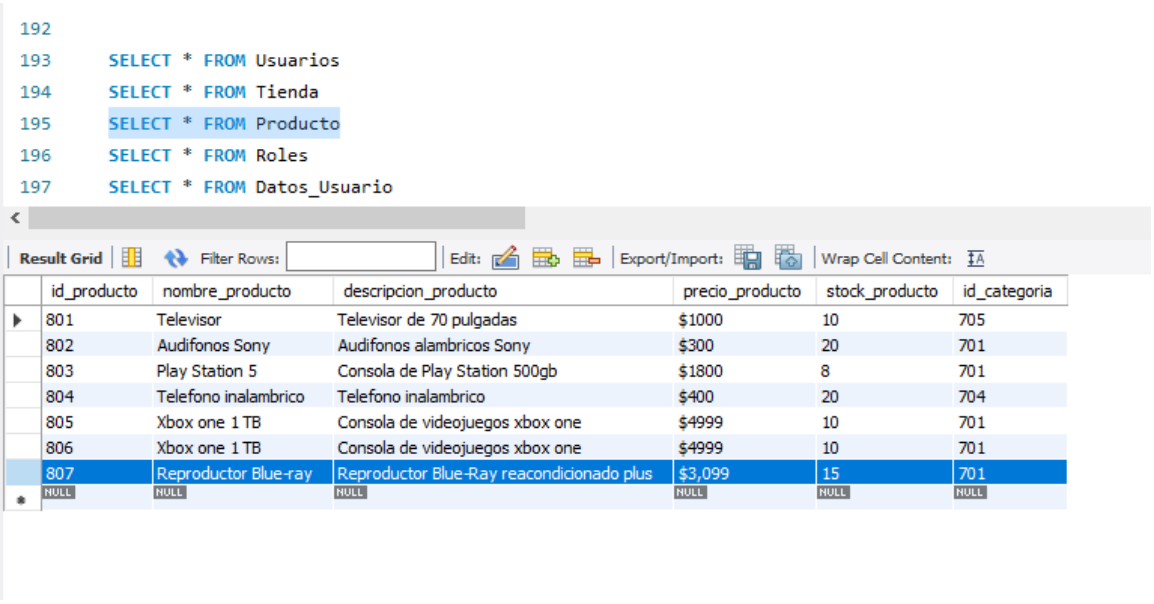
Código de la inserción desde el servidor a la base de datos.

```
//ALTA NUEVO PRODUCTO

ruta.post('/alta', (req,res)=>{
    const {id_producto, nombre_producto, descripcion_producto,
precio_producto, stock_producto, id_categoria}=req.body;
    let query = "INSERT INTO
`proyectobd`.`Producto`(`id_producto`,`nombre_producto`,`descripcion_prod
ucto`,`precio_producto`,`stock_producto`, `id_categoria`) VALUES
('"+id_producto+"','"+nombre_producto+"','"+descripcion_producto+"',
'"+precio_producto+"', '"+stock_producto+"', '"+id_categoria+"');"
    consulta.query(query,(err,rows)=>{
        if(!err) res.json("Nuevo producto registrado")
        else
            console.error(err)
    })
})
```

Inserción del producto en POSTMAN por medio de la ruta marcada para el alta y especificando la información requerida en cada uno de los campos de la tabla.



Comprobación del registro del nuevo producto en la base de datos.



id_producto	nombre_producto	descripcion_producto	precio_producto	stock_producto	id_categoria
801	Televisor	Televisor de 70 pulgadas	$1000	10	705
802	Audifonos Sony	Audifonos alambricos Sony	$300	20	701
803	Play Station 5	Consola de Play Station 500gb	$1800	8	701
804	Telefono inalambrico	Telefono inalambrico	$400	20	704
805	Xbox one 1 TB	Consola de videojuegos xbox one	$4999	10	701
806	Xbox one 1 TB	Consola de videojuegos xbox one	$4999	10	701
807	Reproductor Blue-ray	Reproductor Blue-Ray reacondicionado plus	$3,099	15	701
NULL	NULL	NULL	NULL	NULL	NULL

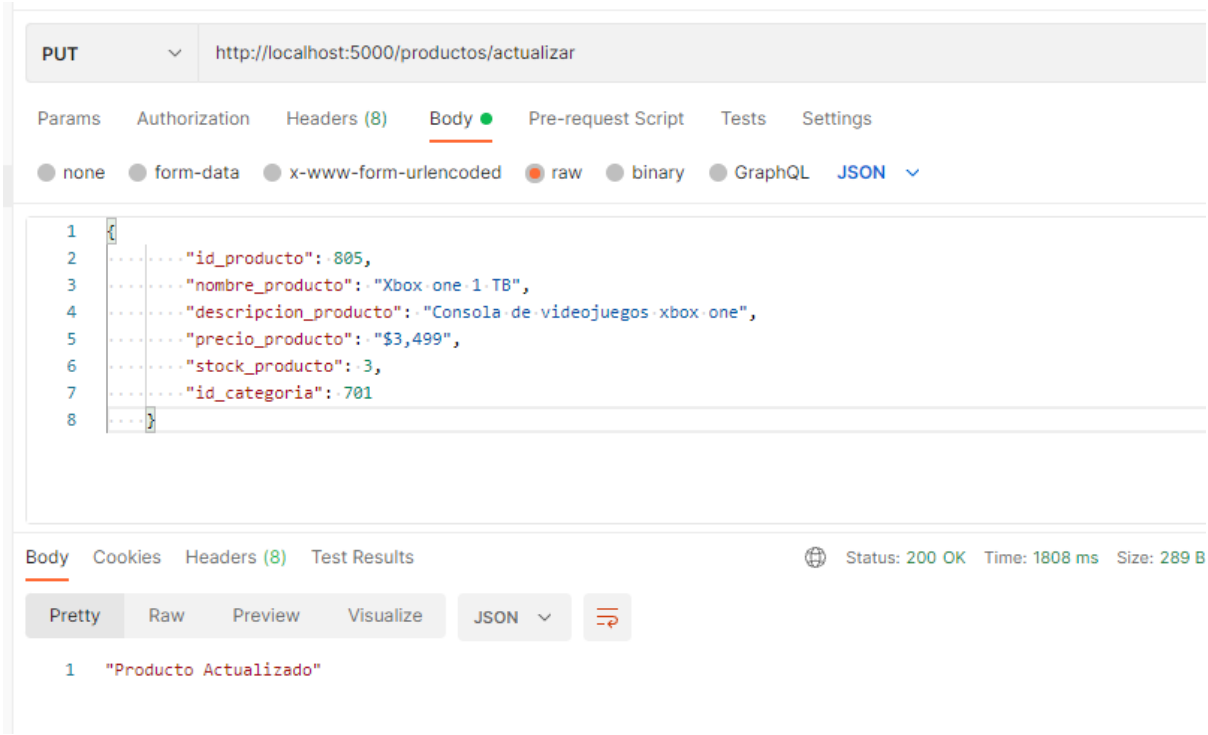
ACTUALIZACIÓN DE UN PRODUCTO (PUT)

Ahora actualizaremos el producto 805 imaginando que su precio y stock han reducido.

Código de la actualización del producto desde el servidor a la base de datos.

```
//ACTUALIZACION PRODUCTO YA REGISTRADO

ruta.put('/actualizar', (req,res)=>{
    const {id_producto,nombre_producto,descripcion_producto,
precio_producto, stock_producto, id_categoria}=req.body;
    let query = "UPDATE `proyectobd`.`Producto` SET `id_producto` =
'"+id_producto+"', `nombre_producto` = '"+nombre_producto+"',
`descripcion_producto` = '"+descripcion_producto+"', `precio_producto` =
'"+precio_producto+"', `stock_producto` = '"+stock_producto+"',
`id_categoria` = '"+id_categoria+"' WHERE (`id_producto` =
'"+id_producto+"')"
    consulta.query(query,(err,rows)=>{
        if(!err) res.json("Producto Actualizado")
        else
            console.error(err)
    })
})
```

Inserción de los campos actualizados del producto en POSTMAN por medio de la ruta. Supongamos que el producto 805 baja de precio y ya quedan pocas unidades de el. Para ello podemos modificarlo de esta manera.



Comprobación de la modificación del precio y stock del producto en la base de datos.

```
191
192
193     SELECT * FROM Usuarios
194     SELECT * FROM Tienda
195     SELECT * FROM Producto
196     SELECT * FROM Roles
197     SELECT * FROM Datos_Usuario
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id_producto	nombre_producto	descripcion_producto	precio_producto	stock_producto	id_categoria
801	Televisor	Televisor de 70 pulgadas	$1000	10	705
802	Audifonos Sony	Audifonos alambricos Sony	$300	20	701
803	Play Station 5	Consola de Play Station 500gb	$1800	8	701
804	Telefono inalambrico	Telefono inalambrico	$400	20	704
805	Xbox one 1 TB	Consola de videojuegos xbox one	$3,499	3	701
806	Xbox one 1 TB	Consola de videojuegos xbox one	$4999	10	701
807	Reproductor Blue-ray	Reproductor Blue-Ray reacondicionado plus	$3,099	15	701
NULL	NULL	NULL	NULL	NULL	NULL

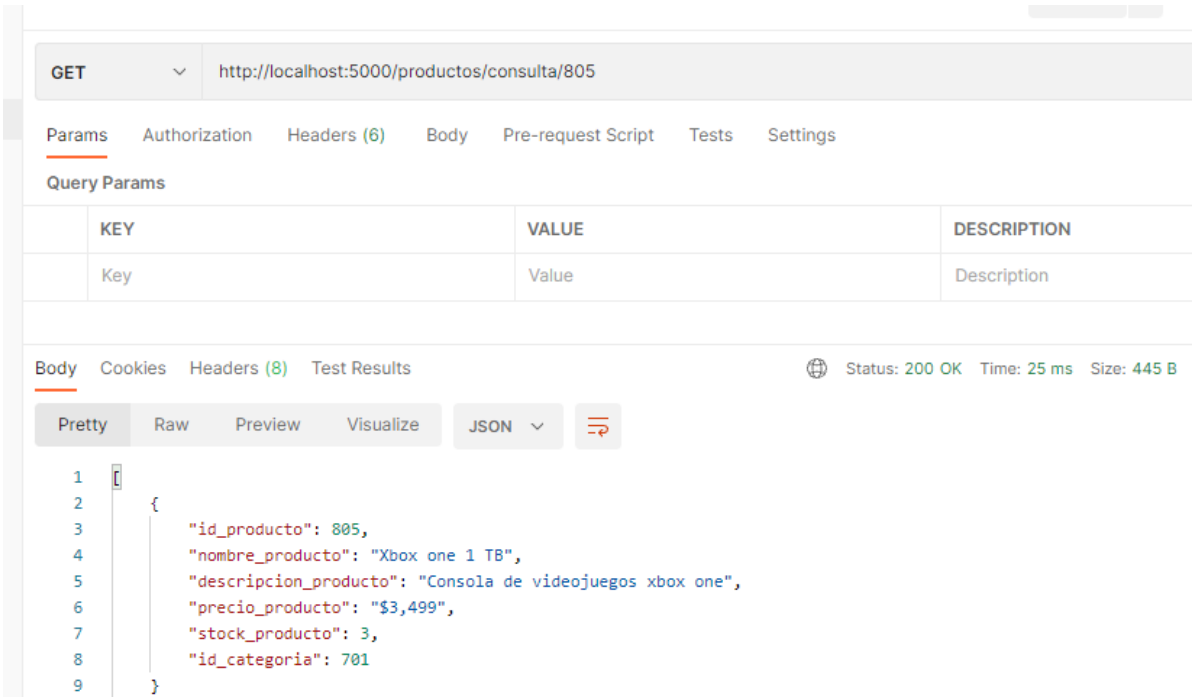
CONSULTA DE UN PRODUCTO POR ID (GET)

Haciendo referencia al paso anterior en donde actualizamos el precio y stock de un producto, de igual manera podemos consultarlo por medio de POSTMAN, esto por medio de la ruta de consulta y posteriormente el ID del producto modificado.

Código de la consulta de un producto por su id.

```
//CONSULTA PRODUCTO POR ID

ruta.get('/consulta/:id',(req,res)=>{
    const {id}=req.params;
    let query = "Select * from Producto where id_producto=?"
    consulta.query(query,[id],(err,rows)=>{
        if(!err) res.json(rows)
        else
            console.error(err)
    })
})
```

Comprobación de la modificación del producto.



BAJA DE UN PRODUCTO POR ID (DELETE)

Si hacemos un análisis a los productos de la tabla, podemos identificar que tenemos un registro duplicado. En este caso el producto 805 y 806.

```
93    SELECT * FROM Usuarios
94    SELECT * FROM Tienda
95    SELECT * FROM Producto
96    SELECT * FROM Roles
97    SELECT * FROM Datos_Usuario
```

esult Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

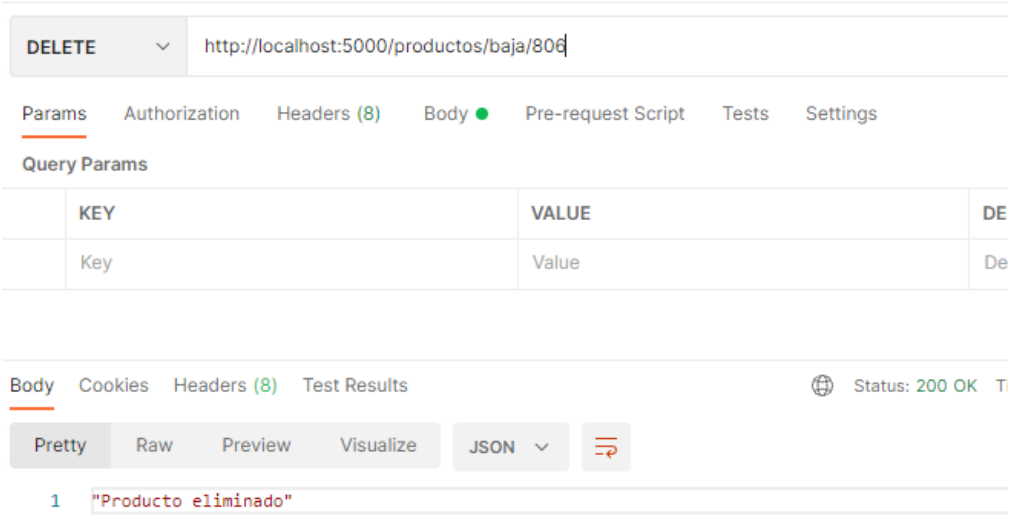
id_producto	nombre_producto	descripcion_producto	precio_producto	stock_producto	id_categoria
801	Televisor	Televisor de 70 pulgadas	$1000	10	705
802	Audifonos Sony	Audifonos alambricos Sony	$300	20	701
803	Play Station 5	Consola de Play Station 500gb	$1800	8	701
804	Telefono inalambrico	Telefono inalambrico	$400	20	704
805	Xbox one 1 TB	Consola de videojuegos xbox one	$3,499	3	701
806	Xbox one 1 TB	Consola de videojuegos xbox one	$4999	10	701
807	Reproductor Blue-ray	Reproductor Blue-Ray reacondicionado plus	$3,099	15	701
NULL	NULL	NULL	NULL	NULL	NULL

Código de la baja de un producto por medio de su ID.

```
//BAJA PRODUCTO POR ID

ruta.delete('/baja/:id', (req,res)=>{
    const {id}=req.params;
```

```
    let query = "Delete from Producto where id_producto=?"
    consulta.query(query,[id],(err,rows)=>{
        if(!err) res.json("Producto eliminado")
        else
            console.error(err)
    })
})
```

Realizamos la eliminación del producto duplicado por medio de POSTMAN indicando la ruta de eliminación y el ID del producto a eliminar, que en este caso es el producto 806.



Comprobación de la baja del producto 806 desde la base de datos.

```
192
193     SELECT * FROM Usuarios
194     SELECT * FROM Tienda
195     SELECT * FROM Producto
196     SELECT * FROM Roles
197     SELECT * FROM Datos_Usuario
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id_producto	nombre_producto	descripcion_producto	precio_producto	stock_producto	id_categoria
801	Televisor	Televisor de 70 pulgadas	$1000	10	705
802	Audifonos Sony	Audifonos alambricos Sony	$300	20	701
803	Play Station 5	Consola de Play Station 500gb	$1800	8	701
804	Telefono inalambrico	Telefono inalambrico	$400	20	704
805	Xbox one 1 TB	Consola de videojuegos xbox one	$3,499	3	701
807	Reproductor Blue-ray	Reproductor Blue-Ray reacondicionado plus	$3,099	15	701
NULL	NULL	NULL	NULL	NULL	NULL

Código completo del CRUD en el archivo PRODUCTOS.JS

```
const {Router} = require ('express');
const consulta = require ('../config/conexionbd');
const ruta = Router();

//CONSULTA PRODUCTO POR ID

ruta.get('/consulta/:id',(req,res)=>{
    const {id}=req.params;
    let query = "Select * from Producto where id_producto=?"
    consulta.query(query,[id],(err,rows)=>{
        if(!err) res.json(rows)
        else
            console.error(err)
    })
})


//BAJA PRODUCTO POR ID

ruta.delete('/baja/:id', (req,res)=>{
    const {id}=req.params;
    let query = "Delete from Producto where id_producto=?"
    consulta.query(query,[id],(err,rows)=>{
        if(!err) res.json("Producto eliminado")
        else
            console.error(err)
    })
})


//ALTA NUEVO PRODUCTO

ruta.post('/alta', (req,res)=>{
    const {id_producto, nombre_producto, descripcion_producto,
precio_producto, stock_producto, id_categoria}=req.body;
    let query = "INSERT INTO
`proyectobd`.`Producto`(`id_producto`,`nombre_producto`,`descripcion_prod
ucto`,`precio_producto`,`stock_producto`, `id_categoria`) VALUES
('"+id_producto+"','"+nombre_producto+"','"+descripcion_producto+"',
'"+precio_producto+"', '"+stock_producto+"', '"+id_categoria+"');"
    consulta.query(query,(err,rows)=>{
        if(!err) res.json("Nuevo producto registrado")
        else
            console.error(err)
    })
})

//ACTUALIZACION PRODUCTO YA REGISTRADO

ruta.put('/actualizar', (req,res)=>{
    const {id_producto,nombre_producto,descripcion_producto,
precio_producto, stock_producto, id_categoria}=req.body;
    let query = "UPDATE `proyectobd`.`Producto` SET `id_producto` =
'"+id_producto+"', `nombre_producto` = '"+nombre_producto+"',
`descripcion_producto` = '"+descripcion_producto+"', `precio_producto` =
```

```
'"+precio_producto+"', `stock_producto` = '"+stock_producto+"',
`id_categoria` = '"+id_categoria+"' WHERE (`id_producto` =
'"+id_producto+"')"
    consulta.query(query,(err,rows)=>{
        if(!err) res.json("Producto Actualizado")
        else
            console.error(err)
    })
})


module.exports = ruta;
```

Peticiones en POSTMAN